

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-329022

(43)Date of publication of application : 13.12.1996

(51)Int.Cl.

G06F 15/16
G06F 9/46

(21)Application number : 07-131511

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 30.05.1995

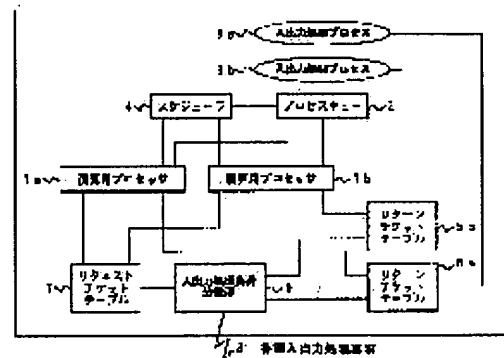
(72)Inventor : TOMOTA MASANORI

(54) INPUT/OUTPUT PROCESS LOAD DECENTRALIZATION CONTROL SYSTEM FOR MULTIPROCESSOR SYSTEM

(57)Abstract:

PURPOSE: To allow an optimum processor among processors perform an interruption process such as the generation, completion, etc., of an input/output process from an input/output device by registering the input/output process in a return ticket table and decentralizing the load of the input/output process to the schedule function of a scheduler.

CONSTITUTION: For example, if an interruption for an input/output process request based upon the input/output process, etc., of the input/output device is generated an input/output process load decentralization part 6 refers to a request ticket table 7 and registers the input/output process in one of return tables 5a and 5b. The request table 7 is so set previously that the input/output process is decentralized equally to the return ticket tables 5a and 5b. Consequently, the interruption processing for the generation, completion, etc., of the input/output process from the input/output device can be performed by an optimum processor between the processors 1a and 1b.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the radial transfer load-distribution control system of the multiprocessor system equipped with the scheduler which is the multiprocessor system which consists of two or more processors, and carries out the schedule of the treatment process to the processor of either of the processors of these plurality according to a predetermined regulation Two or more radial transfer processes of performing various radial transfer which a schedule is carried out to the processor of either of two or more above-mentioned processors with the above-mentioned scheduler, and is demanded from an I/O device, The request ticket table showing whether which radial transfer process in two or more above-mentioned radial transfer processes performs radial transfer demanded from the predetermined I/O device, The return ticket table on which the radial transfer which it is prepared corresponding to each above-mentioned radial transfer process, and the above-mentioned radial transfer process should process, and by which the demand was carried out [above-mentioned] is registered, When the above-mentioned radial transfer is required, the above-mentioned request ticket table is referred to. A radial transfer load-distribution means to register this radial transfer into the return ticket table shown on this request ticket table is provided. The radial transfer load-distribution control system of the multiprocessor system characterized by distributing the load of the radial transfer demanded from an I/O device by the schedule function of a scheduler.

[Claim 2] The average-answering-time table which stores the average answering time of the radial transfer which sets up beforehand the processor which processes two or more above-mentioned radial transfer processes of each, and is passed from two or more above-mentioned processors of each per processor, The count table of radial transfer which stores the count of the radial transfer by which the demand was carried out [above-mentioned] per I/O device is prepared. For the above-mentioned radial transfer load-distribution means, when the value of the largest processor of the average answering time further stored in the above-mentioned average-answering-time table is beyond a predetermined reference value With reference to the above-mentioned request ticket table and the count table of radial transfer, an I/O device with most counts of radial transfer is detected out of the I/O device which this processor processes. It has a means to update the above-mentioned request ticket table so that the radial transfer of this I/O device may be assigned to the smallest processor of average answering time. The radial transfer load-distribution control system of the multiprocessor system according to claim 1 characterized by distributing the load of the radial transfer demanded from an I/O device by the distribution function of a radial transfer load-distribution means.

[Claim 3] Furthermore, a means to detect modification of a system configuration including the processor under operation of the above-mentioned multiprocessor system, and the change in an I/O device, The detection result by this means is followed. The above-mentioned radial transfer process, a request ticket table, [when a means to fluctuate a return ticket table, an average-answering-time table, and the count table of radial transfer is provided and a system configuration is changed during system operation] The radial transfer load-distribution control system of the multiprocessor system according to claim 1 or 2 characterized by carrying out by continuing the optimal load distribution, without carrying out a halt and reboot of a system.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention is applied to the multiprocessor system which consists of two or more processors, and relates to a suitable radial transfer load-distribution control system.

[0002]

[Description of the Prior Art] In the multiprocessor system which consists of two or more conventional processors, in order that I/O devices, such as external storage, may notify completion of radial transfer etc. to a processor, interruption is notified from the I/O device to one of processors. On the other hand, the processor which received this interruption is processing completion of radial transfer etc. by performing reception and this interruption processing for parameters, such as a priority of interruption, and received data, by interrupting from each I/O device, specifically minding the controller, shifting, and notifying interruption to that processor.

[0003] Moreover, it is answered to interruption of the completion about that radial transfer demand etc. by the processor which published the demand, when the radial transfer demand at this time is a radial transfer demand to a disk unit etc. and is published from a processor side. Since the processor which published this radial transfer demand is considered to already have gained the resource required for the processing relevant to this radial transfer demand etc., if this interruption processing is made to perform to this processor, its probability that subsequent processing can also be performed efficiently is high.

[0004] In the radial transfer which, on the other hand, minded networks, such as transfer data processing, although the radial transfer demand will be received and it will tell a processor when the demand of network radial transfer, i.e., the radial transfer from the outside, is published, in the case of a multiprocessor system, an interruption controller cannot be judged like [in the radial transfer demand to the disk unit which mentioned above whether it would be most efficient to interrupt which processor in two or more processors]. Because, it is because it does not have the criteria which judge whether it is most efficient for an interruption controller side to make this I/O interruption processing perform to which processor (for example, it is not necessary to compete with other processors with few resources which should be gained from now on etc.).

[0005] In the Prior art, the approach of deciding from the beginning the processor which notifies interruption to be a meaning, the approach of notifying interruption to sequence, etc. are mainly adopted. Moreover, when the function to specify the processor which should notify interruption next from a processor side sticks to the interruption controller, it is possible to perform a certain amount of load distribution according to the system operating status of a system. However, when this load distribution is also specified to the next interruption which cannot guess in time when it is published and an interrupt occurs actually, the load profile initiations of the time of setting up interruption and a processor may differ greatly.

[0006] With reference to drawing 11, actuation and the trouble of the radial transfer in the conventional multiprocessor system are shown. Drawing 1111 is drawing showing actuation and the trouble of the radial transfer in the conventional multiprocessor system.

[0007] In the multiprocessor system shown in drawing 11, 21a and 21b show the processor for an operation, respectively, and 23 shows the I/O device. Moreover, it is the interruption controller which determines whether 22 notifies I/O interruption published from I/O device 23 to which processors 21a-21b for an operation.

[0008] Here, the multiprocessor system which processes radial transfer only by the specific processor for an operation shall be considered first, and radial transfer shall be performed only by the processor 21a side for an operation.

[0009] In this case, I/O interruption published from I/O device 23 will be notified only to processor 21a for an operation from the interruption controller 22. For this reason, there is dramatically much I/O interruption published from I/O device 23, and even if it will be in a condition to the extent that it cannot process only by processor 21a for an operation, since the object which the interruption controller 22 notifies is only processor 21a for an operation, it cannot distribute a part of these processings to processor 21b for an operation.

[0010] Next, although the case where the approach of assigning interruption in order to the specific processors 21a-21b for an operation instead of a processor is taken is considered, the following problems occur also in this case.

[0011] For example, when the I/O interrupt of a large quantity occurs in the inside of a short time from the same input/output processor 23, i.e., an I/O device, these interruption is uniformly assigned to the processors 21a-21b for an operation, and is notified to them.

[0012] However, there are many parts which are performing exclusive control by a software lock etc. in radial transfer, and processing to the same I/O device is hardly carried out by juxtaposition at each processor for an operation. Moreover, since delivery of the cache data between the processors for an operation cannot be performed when the exclusion authority by exclusive control moves by the interprocessor for an operation from which plurality differs, the data transfer from a cache to memory and the data transfer from memory to a cache are needed, and effectiveness gets very bad.

[0013] Namely, it is more efficient to be the same processor for an operation and to perform radial transfer, in order to use a cache and exclusive control well, when effectiveness does not improve and I/O of a large quantity occurs for a short time, even if it assigns the radial transfer of the same I/O device uniformly to each processor for an operation. From this, it can be said that the effective load distribution of radial transfer cannot be performed also with the method which assigns interruption in order to the processor for an operation.

[0014]

[Problem(s) to be Solved by the Invention] In the multiprocessor system which consists of two or more conventional processors as mentioned above When generating, termination, etc. of radial transfer by interruption are notified to a processor, From a thing, like there is

no means to guess the situation of that the function of an interruption controller is limited, the load of a processor, the resource which is carrying out current acquisition, etc., etc. When notifying interruption to which processor in two or more processors in a system, it was not able to judge whether processing could be advanced efficiently. Therefore, decentralization of the processing based on the load balance of the whole system, such as aiming at distribution of interruption, without notifying interruption to the processor defined beforehand, or taking processing effectiveness into consideration at all, was not attained.

[0015] This invention aims at offering the radial transfer load-distribution control system of the multiprocessor system which makes the optimal processor perform interruption processing of generating of the radial transfer from an I/O device, completion, etc. out of two or more processors, taking the load balance of the whole system into consideration in the multiprocessor system which is made in view of the above-mentioned actual condition, and consists of two or more processors.

[0016]

[Means for Solving the Problem] This invention is a multiprocessor system which consists of two or more processors. In the radial transfer load-distribution control system of the multiprocessor system equipped with the scheduler which carries out the schedule of the treatment process to the processor of either of the processors of these plurality according to a predetermined regulation Two or more radial transfer processes of performing various radial transfer which a schedule is carried out to the processor of either of two or more above-mentioned processors with the above-mentioned scheduler, and is demanded from an I/O device, The request ticket table showing whether which radial transfer process in two or more above-mentioned radial transfer processes performs radial transfer demanded from the predetermined I/O device, The return ticket table on which the radial transfer which it is prepared corresponding to each above-mentioned radial transfer process, and the above-mentioned radial transfer process should process, and by which the demand was carried out [above-mentioned] is registered, When the above-mentioned radial transfer is required, the above-mentioned request ticket table is referred to. A radial transfer load-distribution means to register this radial transfer into the return ticket table shown on this request ticket table is provided, and it is characterized by distributing the load of the radial transfer demanded from an I/O device by the schedule function of a scheduler.

[0017] Moreover, this invention sets up beforehand the processor which processes two or more above-mentioned radial transfer processes of each. The average-answering-time table which stores the average answering time of the radial transfer passed from two or more above-mentioned processors of each per processor, The count table of radial transfer which stores the count of the radial transfer by which the demand was carried out [above-mentioned] per I/O device is prepared. For the above-mentioned radial transfer load-distribution means, when the value of the largest processor of the average answering time further stored in the above-mentioned average-answering-time table is beyond a predetermined reference value With reference to the above-mentioned request ticket table and the count table of radial transfer, an I/O device with most counts of radial transfer is detected out of the I/O device which this processor processes. It has a means to update the above-mentioned request ticket table so that the radial transfer of this I/O device may be assigned to the smallest processor of average answering time, and it is characterized by distributing the load of the radial transfer demanded from an I/O device by the distribution function of a radial transfer load-distribution means.

[0018] Moreover, a means by which this invention detects modification of a system configuration including the processor under operation of the above-mentioned multiprocessor system, and the change in an I/O device further, The detection result by this means is followed. The above-mentioned radial transfer process, a request ticket table, [when a means to fluctuate a return ticket table, an average-answering-time table, and the count table of radial transfer is provided and a system configuration is changed during system operation] It is characterized by carrying out by continuing the optimal load distribution, without carrying out a halt and reboot of a system.

[0019]

[Function] According to this invention, when the various radial transfer based on radial transfer completion of an I/O device etc. is required, a radial transfer load-distribution means registers this radial transfer into the return ticket table shown on this request ticket table with reference to a request ticket table first.

[0020] On the other hand, according to a predetermined regulation, the schedule of the radial transfer process is carried out to the processor of either of two or more processors by the scheduler, and it is performed. The radial transfer process which the schedule was carried out to one of these processors, and was performed performs radial transfer registered into this return ticket table with reference to the return ticket table prepared corresponding to each.

[0021] Since the schedule of the radial transfer process is carried out to the optimal processor by the schedule function of a scheduler, load-distribution control of the radial transfer which took the load balance of the whole system into consideration as a result will be realized. Moreover, since all are registered into the same return ticket table even when the radial transfer of a large quantity is required from the same I/O device for a short period of time, it will perform in the same radial transfer process of the processor subordinate for the same operation, and processing effectiveness is not spoiled.

[0022] Moreover, according to this invention, the average answering time of the radial transfer passed from two or more processors of each is stored in the average-answering-time table per processor. Moreover, the count of the radial transfer demanded from each I/O device is stored in the count table of radial transfer per I/O device.

[0023] In this case, the processor which processes each radial transfer process is set up beforehand, and it is not based on the schedule function of a scheduler, but the following procedures realize load-distribution control of radial transfer.

[0024] That is, when radial transfer is required from an I/O device etc., a radial transfer load-distribution means judges whether with reference to the request ticket table and the average-answering-time table, the average answering time of the processor which performs radial transfer of this I/O device has become beyond the predetermined reference value.

[0025] For example, when the average answering time of the radial transfer by this processor is max in the value stored in the average-answering-time table and it is the value beyond predetermined twice of the average of the average answering time of the radial transfer by other processors, it is carrying out to beyond a predetermined reference value etc.

[0026] Thus, as a result of being judged, when an average response time is carried out to beyond a predetermined reference value, with reference to the count table of radial transfer, it judges shortly whether most counts in the radial transfer which this processor performs of the radial transfer by this I/O device are performed.

[0027] When it judges that it is carried out most mostly in the radial transfer which this processor performs by the count of the radial transfer by this I/O device here, a request ticket table is updated so that other processors and a processor with the concrete smallest average answering time of radial transfer may perform radial transfer by this I/O device.

[0028] And a radial transfer load-distribution means registers this radial transfer into the return ticket table shown on the request ticket table. Thus, load-distribution control of the radial transfer according to the system operating status of a system will be realized by updating

a request ticket table dynamically according to the load of each processor.

[0029] Moreover, according to this invention, modification of the system configuration in which a detection means includes the processor under operation of a multiprocessor system and the change in an I/O device is detected, and increase and decrease of a means fluctuate a radial transfer process, a request ticket table, a return ticket table, an average-answering-time table, and the count table of radial transfer according to this detection result.

[0030] It makes it possible to carry out by continuing the optimal load distribution, without carrying out a halt and reboot of a system by this, when a system configuration is changed during system operation.

[0031]

[Example] With reference to a drawing, the example of this invention is explained below. First, the 1st example of this invention is explained with reference to drawing 1 thru/or drawing 4. Drawing 1 is the outline block diagram of the multiprocessor system which comes to apply the radial transfer load-distribution control system concerning the 1st example.

[0032] A process is processed by two or more processors 1a-1b for an operation as the multiprocessor system concerning this example is shown in drawing 1. Moreover, the process performed by these processors 1a-1b for an operation is stored in the process queue 2, and will be in the condition of the waiting for activation. And only the same number as the processors 1a-1b for an operation sets up the radial transfer processes 3a-3b which perform various radial transfer.

[0033] A scheduler 4 carries out the schedule of the process which is stored in the process queue 2 and serves as waiting for activation to either of the processors 1a-1b for an operation. It is carried out to the processor for an operation judged that this schedule is the the best for performing this processing each time according to the system operating status of the processors 1a-1b for an operation etc.

[0034] the return ticket tables 5a-5b -- the radial transfer processes 3a-3b -- it is alike, respectively, and it corresponds, and it is prepared and the radial transfer which these radial transfer processes 3a-3b should process is registered.

[0035] Moreover, the radial transfer load-distribution section 6 registers the demand of radial transfer into either of the return ticket tables 5a-5b according to setting out of the request ticket table 7. This request ticket table 7 shows whether the demanded radial transfer should be registered into which return ticket tables 5a-5b per I/O device.

[0036] Supposing the interrupt of the radial transfer demand based on radial transfer completion of now, for example, an I/O device, etc. occurs, the radial transfer load-distribution section 6 will register this radial transfer into either of the return ticket tables 5a-5b with reference to the request ticket table 7 first.

[0037] For example, as shown in drawing 2, when the request ticket table 7 is set up, the radial transfer of the I/O device of 1 and 3 is registered into the return ticket table recognized by the number of 0, and an I/O device number is registered into the return ticket table on which, as for the radial transfer of the I/O device of 2, an I/O device number is recognized by the number of 1.

[0038] Moreover, drawing 3 is the conceptual diagram showing the return ticket table recognized by the number of 0, and the I/O device number shows the condition that the radial transfer of the I/O device of 1 and 3 is registered, by the radial transfer load-distribution section 6.

[0039] In addition, this request ticket table 7 is beforehand set up, as radial transfer distributed in the return ticket tables 5a-5b at homogeneity. On the other hand, the radial transfer processes 3a-3b are not performed by the specific processor for an operation, and the schedule of them is carried out to the processors 1a-1b for an operation judged to be the optimal according to system operating status etc. by the schedule function of a scheduler 4, and they are performed by it. The radial transfer processes 3a-3b which the schedule was carried out to these processors 1a-1b for an operation judged to be the optimal, and were performed. The return ticket tables 5a-5b prepared corresponding to each are referred to (radial transfer process 3a refers to the return ticket table 5b for return ticket table 5a and radial transfer process 3b, respectively.). Radial transfer registered into the return ticket tables 5a-5b is performed.

[0040] These radial transfer processes 3a-3b confirm again whether the radial transfer of a processor limited is registered with reference to the return ticket tables 5a-5b, when the radial transfer under activation is completed. And when the radial transfer of a processor limited is registered, radial transfer of the processor limited is performed succeedingly. By this, it will be processed in the same operation processor subordinate's same radial transfer process from the same I/O device for a short period of time succeeding the case where the radial transfer of a large quantity occurs.

[0041] In addition, if the radial transfer of a processor limited will be in the condition of not registering with the return ticket tables 5a-5b, it will end and the reclosing of the radial transfer processes 3a-3b will be carried out to the process queue 2. Therefore, in case a schedule is carried out to an operation processor next, a schedule will be carried out to the processor for an operation judged to be the optimal at the event.

[0042] Processing effectiveness when load-distribution control of the radial transfer in consideration of the load balance of the whole system is realized by this and the radial transfer of a large quantity occurs from the same I/O device for a short period of time is not spoiled.

[0043] Actuation of the 1st example is explained with reference to drawing 4. Drawing 4 is a flow chart for explaining actuation of the 1st example. If the radial transfer demand based on radial transfer completion of an I/O device etc. occurs (step A1 of drawing 4), the radial transfer distribution section 6 will register this radial transfer into the return tables 5a-5b shown in the request ticket table 7 with reference to the request ticket table 7 (step A2 of drawing 4).

[0044] On the other hand, a scheduler 4 carries out the schedule of the radial transfer processes 3a-3b registered into the process queue 2 to either of the processors 1a-1b for an operation according to system operating status etc. (step A3 of drawing 4).

[0045] And the radial transfer processes 3a-3b by which the schedule was carried out to either of these processors 1a-1b for an operation perform radial transfer registered into the return ticket tables 5a-5b (step A4 of drawing 4).

[0046] Load-distribution control of the radial transfer which took the load balance of the whole system into consideration by this is realized. Next, the 2nd example of this invention is explained with reference to drawing 5 thru/or drawing 9.

[0047] Drawing 5 is an outline block diagram about that of the multiprocessor system which comes to apply the radial transfer load-distribution control system concerning the 2nd example. In addition to the configuration of the multiprocessor system of the 1st example, the multiprocessor system concerning this example forms the average-answering-time count table 9 and 10 of radial transfer further.

[0048] this average-answering-time table 9 -- the processors 1a-1b for an operation -- respectively -- since -- the average answering time of the various radial transfer passed is stored. Moreover, the count table 10 of radial transfer stores the count of the radial transfer demanded from each I/O device per I/O device.

[0049] this example -- the radial transfer processes 3a-3b -- making the processors 1a-1b for an operation which process each connect with

the specific processors 1a-1b for an operation beforehand, a scheduler 4 carries out the schedule of the radial transfer processes 3a-3b stored in the process queue 2 to the processors 1a-1b for an operation beforehand specified by setting out.
 [0050] Supposing the interrupt of the radial transfer demand based on radial transfer completion of now, for example, an I/O device, etc. occurs First, the radial transfer load-distribution section 6 refers to the request ticket table 7. Whether it is what should register the radial transfer of this I/O device into any of the return ticket tables 5a-5b That is, it judges whether it is the thing which any of the processors 1a-1b for an operation should be made to perform (in this example, since the processor for an operation which performs a radial transfer process is connected statically, selection of a return ticket table turns into selection of the processor for an operation as it is).

[0051] It judges whether by this decision, if the return ticket table which should be registered, i.e., the processor for an operation which performs processing, becomes clear, with reference to the average-answering-time table, the average answering time of this processor for an operation is a value beyond predetermined criteria to the average of the average answering time of other processors for an operation this time.

[0052] And when it becomes clear that it is a value beyond predetermined criteria as a result of this judgment, with reference to the count table 10 of radial transfer, the input/output request of this I/O device judges further whether it is what has most counts among the I/O device which that processor for an operation processes.

[0053] Here, when judged with there being most counts, the radial transfer load-distribution section 6 updates the request ticket table 7 so that the small processor for an operation of average answering time may be made to process the radial transfer of this I/O device most.

[0054] For example, it recognizes that it is the thing to which an I/O device number makes the operation processor of No. 2 process this radial transfer when the radial transfer demand from the I/O device of 5 occurs and the radial transfer load-distribution section 6 refers to the request ticket table 7 in the condition of being set up as shown to drawing 6 in the request ticket table 7.

[0055] Here, the radial transfer load-distribution section 6 judges whether with reference to the average-answering-time table 10, the response time of this operation processor of No. 2 serves as a value beyond predetermined twice to the average of the average answering time of other operation processors this time.

[0056] For example, if the average-answering-time table 10 shows drawing 7 and makes criteria less than 2 double, the average of the average answering time of operation processors other than No. 2 will be set to $(1200+2400) / 2 = 1800$, and the average answering time of the operation processor of No. 2 will become twice [more than] this value.

[0057] At this time, an I/O device number judges whether there is most radial transfer load-distribution section 6 in the radial transfer which the operation processor of No. 2 processes [the I/O device of 5] further with reference to the count table 10 of radial transfer. For example, when the count table 10 of radial transfer is in the condition shown in drawing 8, as for the input/output processor which the operation processor of No. 2 processes, the request ticket table 7 shows that they are No. 3 and No. 5, and, thereby, it becomes clear that there are most counts of the radial transfer of this input/output processor of No. 5 during the radial transfer of the input/output processor which the operation processor of No. 2 processes.

[0058] At this time, the radial transfer load-distribution section 6 updates the request ticket table 7 so that the small operation processor of average answering time may be made to process most the radial transfer of this input/output processor of No. 5. In this case, since the processor for an operation of No. 0 turns into a small operation processor of average answering time most, the processor for an operation which performs radial transfer of the input/output processor of No. 5 of the request ticket table 7 is updated from No. 2 to No. 0.

[0059] Then, the input / output load distribution section 6 registers this radial transfer into a return ticket table according to the request ticket table 7. Thus, in consideration of the average answering time of each processor for an operation, and the count of radial transfer, load-distribution control of the radial transfer according to the system operating status of a system will be realized by updating the request ticket table dynamically.

[0060] Actuation of the 2nd example is explained with reference to drawing 9. Drawing 9 is a flow chart for explaining actuation of the 2nd example. If the radial transfer demand based on radial transfer completion of an I/O device etc. occurs (step B1 of drawing 9), the radial transfer distribution section 6 will count up the count table 10 of radial transfer first (step B-2 of drawing 9).

[0061] Next, the radial transfer distribution section 6 acquires the processor number for an operation which performs this radial transfer with reference to a request ticket table (step B3 of drawing 9), and acquires the average answering time of this processor number for an operation with reference to the average-answering-time table 9 further (step B4 of drawing 9).

Here, the radial transfer distribution section 6 judges whether this average response time has become to the average of the average response time of other processors for an operation beyond the predetermined reference value (step B5 of drawing 9).

[0062] Here, when judged with it being beyond a reference value, it judges [(Y of step B5 of drawing 9), and] whether the count of generating of radial transfer is the most numerous in a processor for this operation further (step B6 of drawing 9).

[0063] Here, when judged with it being the most numerous, a request ticket table is updated so that the small processor for an operation of average answering time may be made to process most (Y of step B6 of drawing 9), and the radial transfer of this I/O device (step B7 of drawing 9).

[0064] Then, the radial transfer load-distribution section 6 registers this radial transfer into the return table specified on the request ticket table (step B8 of drawing 9).

[0065] On the other hand, a scheduler 4 carries out the schedule of the radial transfer process registered into the process queue 2 to the processor for an operation beforehand specified by setting out (step B9 of drawing 9 R> 9). And the radial transfer process by which the schedule was carried out to this processor for an operation beforehand specified by setting out performs radial transfer registered into the return ticket table (step B10 of drawing 9).

[0066] Load-distribution control of the radial transfer which took the load balance of the whole system into consideration by this, without being based on the schedule function of a scheduler is realized. In addition, as shown in drawing 10, from the equipment installation detecting element 11 further, and modification of a system configuration including the change in the processor for an operation under operation of a multiprocessor system, an I/O device, etc. is detected. If the change in a radial transfer process, a request ticket table, a return ticket table, an average-answering-time table, and the count table of radial transfer is made to carry out dynamically according to this detection result It makes it possible to carry out by continuing the optimal load distribution, without carrying out a halt and reboot of a system, when a system configuration is changed during system operation.

[0067]

[Effect of the Invention] It makes it possible to carry out by continuing the optimal load distribution, without carrying out a halt and reboot of a system, when load-distribution control of the radial transfer in consideration of the load balance of the whole system is realized,

without spoiling processing effectiveness according to this invention as explained in full detail above, and a system configuration is changed during system operation.

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The outline block diagram of the multiprocessor system which comes to apply the radial transfer load-distribution control system concerning the 1st example of this invention.

[Drawing 2] The conceptual diagram showing the request ticket table concerning the 1st example.

[Drawing 3] The conceptual diagram showing the return ticket table concerning the 1st example.

[Drawing 4] The flow chart for explaining actuation of the 1st example.

[Drawing 5] The outline block diagram of the multiprocessor system which comes to apply the radial transfer load-distribution control system concerning the 2nd example of this invention.

[Drawing 6] The conceptual diagram showing the request ticket table concerning the 2nd example.

[Drawing 7] The conceptual diagram showing the average-answering-time table concerning the 2nd example.

[Drawing 8] The conceptual diagram showing the count table of radial transfer concerning the 2nd example.

[Drawing 9] The flow chart for explaining actuation of the 2nd example.

[Drawing 10] The outline block diagram of the multiprocessor system which comes to apply the radial transfer load-distribution control system equipped with the equipment installation detection means.

[Drawing 11] Drawing showing actuation and the trouble of the radial transfer in the conventional multiprocessor system.

[Description of Notations]

1a, 1b [-- A scheduler 5a, 5b / -- A return ticket table, 6 / -- The radial transfer distribution section, 7 / -- A request ticket table, 8 / -- Various radial transfer 9 / -- Average answering time, 10 / -- The count table of radial transfer, 11 / -- Equipment installation detection section.] -- The processor for an operation, 2 -- A process queue, 3a, 3b -- A radial transfer process, 4

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-329022

(43) 公開日 平成8年(1996)12月13日

(51) Int.Cl. ⁹	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 8 0		G 0 6 F 15/16	3 8 0 Z
9/46	3 6 0		9/46	3 6 0 A

審査請求 未請求 請求項の数 3 O L (全 12 頁)

(21) 出願番号 特願平7-131511

(22) 出願日 平成7年(1995)5月30日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 友田 正憲

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

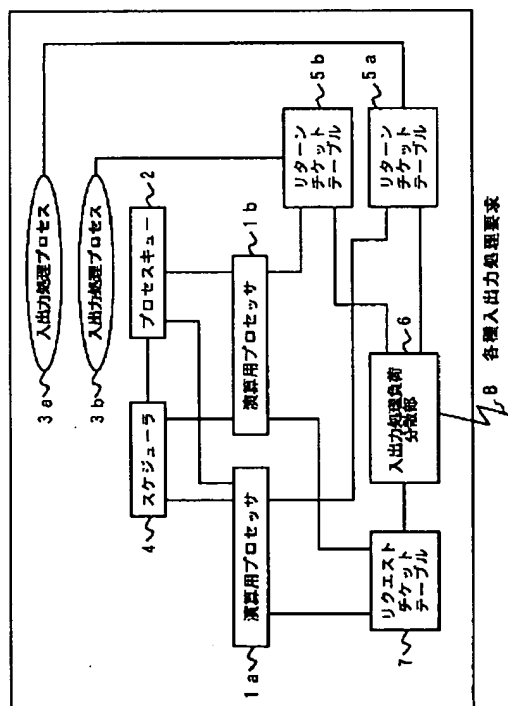
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 マルチプロセッサシステムの入出力処理負荷分散制御方式

(57) 【要約】

【目的】入出力処理の発生及び完了等の割り込み処理を複数のプロセッサの中から最適なプロセッサに実行させる入出力処理負荷分散制御方式を提供する。

【構成】スケジューラ4によりいずれかの演算用プロセッサ1a～1bにスケジュールされ、各種入出力処理を実行する入出力処理プロセス3a～3bと、要求された入出力処理をいずれの入出力処理プロセス3a～3bが実行するのを示すリクエストチケットテーブル7と、上記入出力処理プロセス3a～3bそれぞれに対応して設けられ、上記入出力処理プロセスが処理すべき上記要求された入出力処理が登録されるリターンチケットテーブル5a～5bと、上記入出力処理が要求された際に、上記リクエストチケットテーブルにより示されたリターンチケットテーブルにこの入出力処理を登録する入出力処理負荷分散部6とを具備してなることを特徴とする。



【特許請求の範囲】

【請求項1】 複数のプロセッサからなるマルチプロセッサシステムであって、処理プロセスを所定の規則に従ってこれら複数のプロセッサの中のいずれかのプロセッサにスケジューリングするスケジューラを備えたマルチプロセッサシステムの入出力処理負荷分散制御方式において、

上記スケジューラにより上記複数のプロセッサの中のいずれかのプロセッサにスケジューリングされ、入出力装置から要求される各種入出力処理を実行する複数の入出力処理プロセスと、所定の入出力装置から要求された入出力処理を上記複数の入出力処理プロセスの中のいずれの入出力処理プロセスが実行するのを示すリクエストチケットテーブルと、上記入出力処理プロセスそれぞれに対応して設けられ、上記入出力処理プロセスが処理すべき上記要求された入出力処理が登録されるリターンチケットテーブルと、上記入出力処理が要求された際に、上記リクエストチケットテーブルを参照し、このリクエストチケットテーブルにより示されたリターンチケットテーブルにこの入出力処理を登録する入出力処理負荷分散手段とを具備し、入出力装置から要求される入出力処理の負荷をスケジューラのスケジューリング機能により分散することを特徴とするマルチプロセッサシステムの入出力処理負荷分散制御方式。

【請求項2】 上記複数の入出力処理プロセスそれぞれを処理するプロセッサを予め設定しておき、上記複数のプロセッサそれぞれから渡される入出力処理の平均応答時間をプロセッサ単位に格納する平均応答時間テーブルと、上記要求された入出力処理の回数を入出力装置単位に格納する入出力処理回数テーブルとを設け、上記入出力処理負荷分散手段に、さらに上記平均応答時間テーブルに格納された平均応答時間の最も大きいプロセッサの値が所定の基準値以上であるときに、上記リクエストチケットテーブル及び入出力処理回数テーブルを参照し、このプロセッサが処理する入出力装置の中から最も入出力処理の回数の多い入出力装置を検出して、この入出力装置の入出力処理を平均応答時間の最も小さいプロセッサに割り振られるように上記リクエストチケットテーブルを更新する手段を備え、入出力装置から要求される入出力処理の負荷を入出力処理負荷分散手段の振分機能により分散することを特徴とする請求項1記載のマルチプロセッサシステムの入出力処理負荷分散制御方式。

【請求項3】 さらに、上記マルチプロセッサシステムの稼働中におけるプロセッサ及び入出力装置の増減を含むシステム構成の変更を検知する手段と、この手段による検知結果に従って、上記入出力処理プロセス、リクエストチケットテーブル、リターンチケットテーブル、平均応答時間テーブル、及び入出力処理回数テーブルを増減する手段とを具備し、システム稼働中にシステム構成が変更された際においても、システムの停止及び再起動

を実施することなく最適な負荷分散を継続して行うことを特徴とする請求項1又は2記載のマルチプロセッサシステムの入出力処理負荷分散制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数のプロセッサからなるマルチプロセッサシステムに適用して好適な入出力処理負荷分散制御方式に関する。

【0002】

【従来の技術】従来の複数のプロセッサからなるマルチプロセッサシステムにおいては、外部記憶装置等の入出力装置が入出力処理の完了等をプロセッサに通知するために、入出力装置からいずれかのプロセッサに対して割り込みを通知している。具体的には、各入出力装置から割り込みコントローラを介していずれかのプロセッサに割り込みを通知し、一方、この割り込みを受信したプロセッサは、割り込みの優先度や受信データ等のパラメータを受け取り、この割り込み処理を行うことにより入出力処理の完了等の処理を行っている。

【0003】また、このときの入出力処理要求が、ディスク装置等への入出力処理要求であって、プロセッサ側から発行されたものである場合には、その入出力処理要求についての完了等の割り込みは、要求を発行したプロセッサに返答されるようになっている。この入出力処理要求を発行したプロセッサは、この入出力処理要求に関連した処理に必要な資源等を既に獲得していると考えられるため、この割り込み処理をこのプロセッサに行わせれば、その後の処理も効率良く行うことができる蓋然性が高い。

【0004】一方、転送データ処理等のネットワークを介した入出力処理では、ネットワーク入出力処理、即ち、外部からの入出力処理の要求が発行された場合、その入出力処理要求を受信してプロセッサに伝えることになるが、マルチプロセッサシステムの場合、割り込みコントローラは、複数のプロセッサの中のいずれのプロセッサに割り込むのが最も効率が良いかを上述したディスク装置等への入出力処理要求の場合のように判断することができない。なぜならば、割り込みコントローラ側には、いずれのプロセッサに、この入出力割り込み処理を行わせるのが最も効率が良いか（例えば、これから獲得すべき資源が最も少ない、他のプロセッサと競合しなくて済む等）を判断する基準を持たないためである。

【0005】従来の技術では、割り込みを通知するプロセッサを最初から一意に決めておく方法や、順番に割り込みを通知していく方法等が主に採用されている。また、割り込みコントローラに対して、次に割り込みを通知すべきプロセッサをプロセッサ側から指定する機能がついている場合には、システムの稼働状況に応じた、ある程度の負荷分散を行うことが可能である。しかし、この負荷分散も、いつ発行されるのか時間的に推測できな

い次の割り込みに対して指定するものであり、実際に割り込みが発生した時点では、割り込みを設定した時とプロセッサの負荷状況が大きく異なっている場合がある。

【0006】図11を参照して従来のマルチプロセッサシステムでの入出力処理の動作とその問題点を示す。図11は従来のマルチプロセッサシステムでの入出力処理の動作とその問題点を示す図である。

【0007】図11に示すマルチプロセッサシステムにおいて、21a、21bはそれぞれ演算用プロセッサを示しており、23は入出力装置を示している。また、22は入出力装置23から発行された入出力割り込みをいずれの演算用プロセッサ21a～21bに通知するのかを決定する割り込みコントローラである。

【0008】ここでは、まず、入出力処理を特定の演算用プロセッサでのみ処理するマルチプロセッサシステムを考え、演算用プロセッサ21a側でのみ入出力処理を行うものとする。

【0009】この場合、入出力装置23から発行された入出力割り込みは、割り込みコントローラ22から演算用プロセッサ21aにのみ通知されることになる。このため、入出力装置23から発行された入出力割り込みが非常に多く、演算用プロセッサ21aのみでは処理しきれないほどの状態になっても、割り込みコントローラ22が通知する対象は演算用プロセッサ21aのみであるために、演算用プロセッサ21bにこれらの処理の一部を分散することができない。

【0010】次に、割り込みを特定のプロセッサではなく、演算用プロセッサ21a～21bに順番に割り振る方法を取った場合を考えるが、この場合においても、以下のような問題が発生する。

【0011】例えば、同一の入出力処理装置、即ち、入出力装置23から短時間のうちに大量の入出力割り込みが発生したような場合、これらの割り込みは、演算用プロセッサ21a～21bに均等に割り振られて通知される。

【0012】しかし、入出力処理では、ソフトウェアロック等による排他制御を行っている部分が多く、各演算用プロセッサで同一の入出力装置に対する処理が並列に実施されることはほとんどない。また、複数の異なる演算用プロセッサ間で排他制御による排他権限が移動していく場合、演算用プロセッサ相互間でのキャッシュデータの受け渡しができないために、キャッシュからメモリへのデータ転送、及びメモリからキャッシュへのデータ転送を必要とし、効率が非常に悪くなる。

【0013】即ち、同一の入出力装置の入出力処理を均等に各演算用プロセッサに割り振っても効率が向上することはなく、短時間に大量の入出力が発生した場合には、キャッシュや排他制御をうまく使用するために、同一の演算用プロセッサで、入出力処理を行った方が効率がよい。このことから、割り込みを演算用プロセッサに

順番に割り振る方式によっても、入出力処理の効果的な負荷分散を行うことはできないといえる。

【0014】

【発明が解決しようとする課題】上述したように、従来の複数のプロセッサからなるマルチプロセッサシステムにおいては、割り込みによる入出力処理の発生や終了等をプロセッサに通知する場合、割り込みコントローラの機能が限定されていることや、プロセッサの負荷及び現在獲得している資源などの状況を推測する手段がない等のことから、システム内の複数のプロセッサの中のいずれのプロセッサに割り込みを通知すれば処理を効率よく進めることができるかを判定できなかった。従って、予め定められたプロセッサに割り込みを通知する、又は処理効率をまったく考慮せずに割り込みの分散を図る等、システム全体の負荷バランスを踏まえた処理の分散化が図られていなかった。

【0015】本発明は、上記実情に鑑みてなされたものであり、複数のプロセッサからなるマルチプロセッサシステムにおいて、入出力装置からの入出力処理の発生及び完了等の割り込み処理を、システム全体の負荷バランスを考慮しつつ、複数のプロセッサの中から最適なプロセッサに実行させるマルチプロセッサシステムの入出力処理負荷分散制御方式を提供することを目的とする。

【0016】

【課題を解決するための手段】本発明は、複数のプロセッサからなるマルチプロセッサシステムであって、処理プロセスを所定の規則に従ってこれら複数のプロセッサの中のいずれかのプロセッサにスケジュールするスケジューラを備えたマルチプロセッサシステムの入出力処理負荷分散制御方式において、上記スケジューラにより上記複数のプロセッサの中のいずれかのプロセッサにスケジュールされ、入出力装置から要求される各種入出力処理を実行する複数の入出力処理プロセスと、所定の入出力装置から要求された入出力処理を上記複数の入出力処理プロセスの中のいずれの入出力処理プロセスが実行するのかを示すリクエストチケットテーブルと、上記入出力処理プロセスそれぞれに対応して設けられ、上記入出力処理プロセスが処理すべき上記要求された入出力処理が登録されるリターンチケットテーブルと、上記入出力処理が要求された際に、上記リクエストチケットテーブルを参照し、このリクエストチケットテーブルにより示されたリターンチケットテーブルにこの入出力処理を登録する入出力処理負荷分散手段とを具備し、入出力装置から要求される入出力処理の負荷をスケジューラのスケジュール機能により分散することを特徴とする。

【0017】また、本発明は、上記複数の入出力処理プロセスそれぞれを処理するプロセッサを予め設定しておき、上記複数のプロセッサそれぞれから渡される入出力処理の平均応答時間をプロセッサ単位に格納する平均応答時間テーブルと、上記要求された入出力処理の回数を

入出力装置単位に格納する入出力処理回数テーブルとを設け、上記入出力処理負荷分散手段に、さらに上記平均応答時間テーブルに格納された平均応答時間の最も大きいプロセッサの値が所定の基準値以上であるときに、上記リクエストチケットテーブル及び入出力処理回数テーブルを参照し、このプロセッサが処理する入出力装置の中から最も入出力処理の回数の多い入出力装置を検出して、この入出力装置の入出力処理を平均応答時間の最も小さいプロセッサに割り振られるように上記リクエストチケットテーブルを更新する手段を備え、入出力装置から要求される入出力処理の負荷を入出力処理負荷分散手段の振分機能により分散することを特徴とする。

【0018】また、本発明は、さらに、上記マルチプロセッサシステムの稼働中におけるプロセッサ及び入出力装置の増減を含むシステム構成の変更を検知する手段と、この手段による検知結果に従って、上記入出力処理プロセス、リクエストチケットテーブル、リターンチケットテーブル、平均応答時間テーブル、及び入出力処理回数テーブルを増減する手段とを具備し、システム稼働中にシステム構成が変更された際においても、システムの停止及び再起動を実施することなく最適な負荷分散を継続して行うことを特徴とする。

【0019】

【作用】本発明によれば、入出力装置の入出力処理完了等に基づく各種入出力処理を要求された際に、まず、入出力処理負荷分散手段が、リクエストチケットテーブルを参照し、このリクエストチケットテーブルにより示されたリターンチケットテーブルに、この入出力処理を登録する。

【0020】一方、入出力処理プロセスは、所定の規則に従って、スケジューラにより複数のプロセッサの中のいずれかのプロセッサにスケジュールされて実行される。このいずれかのプロセッサにスケジュールされて実行された入出力処理プロセスは、それぞれに対応して設けられたリターンチケットテーブルを参照し、このリターンチケットテーブルに登録された入出力処理を実行する。

【0021】入出力処理プロセスは、スケジューラのスケジュール機能により最も最適なプロセッサにスケジュールされるため、結果的にシステム全体の負荷バランスを考慮した入出力処理の負荷分散制御が実現されることとなる。また、同一の入出力装置から短期間に大量の入出力処理を要求された場合でも、全て同一のリターンチケットテーブルに登録されるため、同一演算用プロセッサ配下の同一入出力処理プロセスで実行されることになり、処理効率を損なうこともない。

【0022】また、本発明によれば、複数のプロセッサそれぞれから渡される入出力処理の平均応答時間を平均応答時間テーブルにプロセッサ単位に格納しておく。また、各入出力装置から要求された入出力処理の回数を入

出力処理回数テーブルに入出力装置単位に格納しておく。

【0023】この場合、入出力処理プロセスそれぞれを処理するプロセッサを予め設定しておき、スケジューラのスケジュール機能によらず、以下の手順により入出力処理の負荷分散制御を実現する。

【0024】即ち、入出力装置等から入出力処理を要求された際に、入出力処理負荷分散手段は、リクエストチケットテーブル及び平均応答時間テーブルを参照し、この入出力装置の入出力処理を実行するプロセッサの平均応答時間が所定の基準値以上になっていないかを判定する。

【0025】例えば、このプロセッサによる入出力処理の平均応答時間が平均応答時間テーブルに格納された値の中で最大であり、かつ他のプロセッサによる入出力処理の平均応答時間の平均値の所定倍以上の値であるときに、所定の基準値以上とする等である。

【0026】このようにして判定された結果、平均応答時間が所定の基準値以上とされた場合、今度は、この入出力装置による入出力処理の回数が、このプロセッサが行う入出力処理の中で最も多く行われているか否かを入出力処理回数テーブルを参照して判定する。

【0027】ここで、この入出力装置による入出力処理の回数が、このプロセッサが行う入出力処理の中で最も多く行われていると判定された場合には、この入出力装置による入出力処理を、他のプロセッサ、具体的には、最も入出力処理の平均応答時間が小さいプロセッサに実行されるように、リクエストチケットテーブルを更新する。

【0028】そして、入出力処理負荷分散手段は、リクエストチケットテーブルにより示されたリターンチケットテーブルに、この入出力処理を登録する。このように、各プロセッサの負荷に応じて動的にリクエストチケットテーブルを更新することにより、システムの稼働状況に応じた入出力処理の負荷分散制御が実現されることとなる。

【0029】また、本発明によれば、検知手段が、マルチプロセッサシステムの稼働中におけるプロセッサ及び入出力装置の増減を含むシステム構成の変更を検知し、この検知結果に従って、増減手段が、入出力処理プロセス、リクエストチケットテーブル、リターンチケットテーブル、平均応答時間テーブル、及び入出力処理回数テーブルを増減する。

【0030】これにより、システム稼働中にシステム構成が変更された際においても、システムの停止及び再起動を実施することなく最適な負荷分散を継続して行うことを可能とする。

【0031】

【実施例】以下図面を参照して本発明の実施例を説明する。まず、図1乃至図4を参照して本発明の第1実施例

を説明する。図1は、第1実施例に係る入出力処理負荷分散制御方式を適用してなるマルチプロセッサシステムの概略構成図である。

【0032】同実施例に係るマルチプロセッサシステムは、図1に示すように、複数の演算用プロセッサ1a～1bによりプロセスが処理される。また、この演算用プロセッサ1a～1bにて実行されるプロセスは、プロセスキュー2に蓄えられ、実行待ちの状態となる。そして、各種入出力処理を実行する入出力処理プロセス3a～3bを、演算用プロセッサ1a～1bと同じ数だけ設定する。

【0033】スケジューラ4は、プロセスキュー2に蓄えられて実行待ちとなっているプロセスを、演算用プロセッサ1a～1bのいずれかにスケジュールする。このスケジュールは、演算用プロセッサ1a～1bの稼働状況等に応じて、その都度この処理を行うのに最適と判断された演算用プロセッサに対して行われる。

【0034】リターンチケットテーブル5a～5bは、入出力処理プロセス3a～3bそれぞれに対応して設けられ、これら入出力処理プロセス3a～3bが処理すべき入出力処理が登録される。

【0035】また、入出力処理負荷分散部6は、入出力処理の要求を、リクエストチケットテーブル7の設定に従って、リターンチケットテーブル5a～5bのいずれかに登録する。このリクエストチケットテーブル7は、入出力装置単位に、要求された入出力処理をいずれのリターンチケットテーブル5a～5bに登録すべきかを示すものである。

【0036】いま、例えば入出力装置の入出力処理完了等に基づく入出力処理要求の割り込みが発生したとすると、まず、入出力処理負荷分散部6が、リクエストチケットテーブル7を参照し、この入出力処理をリターンチケットテーブル5a～5bのいずれかに登録する。

【0037】例えば、リクエストチケットテーブル7が、図2に示すように設定されていた場合、入出力装置番号が1及び3の入出力装置の入出力処理は、0の番号で認識されるリターンチケットテーブルに登録され、また、入出力装置番号が2の入出力装置の入出力処理は、1の番号で認識されるリターンチケットテーブルに登録される。

【0038】また、図3は、0の番号で認識されるリターンチケットテーブルを示す概念図であり、入出力処理負荷分散部6により入出力装置番号が1及び3の入出力装置の入出力処理が登録されている状態を示している。

【0039】なお、このリクエストチケットテーブル7は、入出力処理がリターンチケットテーブル5a～5bに均一に分散されるように予め設定しておく。一方、入出力処理プロセス3a～3bは、特定の演算用プロセッサで実行されるものではなく、スケジューラ4のスケジュール機能により、稼働状況等に応じて最適と判断され

た演算用プロセッサ1a～1bにスケジュールされ実行される。この最適と判断された演算用プロセッサ1a～1bにスケジュールされ実行された入出力処理プロセス3a～3bは、それぞれに対応して設けられたリターンチケットテーブル5a～5bを参照し（入出力処理プロセス3aはリターンチケットテーブル5a、入出力処理プロセス3bはリターンチケットテーブル5bをそれぞれ参照する。）、リターンチケットテーブル5a～5bに登録された入出力処理を実行する。

【0040】この入出力処理プロセス3a～3bは、実行中の入出力処理が終了した際、再度リターンチケットテーブル5a～5bを参照し、処理待ちの入出力処理が登録されていないかをチェックする。そして、処理待ちの入出力処理が登録されている場合には、引き続きその処理待ちの入出力処理を実行する。これにより、同一の入出力装置から短期間に大量の入出力処理が発生した場合に、連続して同一の演算プロセッサ配下の同一入出力処理プロセスで処理されることになる。

【0041】なお、処理待ちの入出力処理がリターンチケットテーブル5a～5bに登録されていない状態になると、入出力処理プロセス3a～3bは終了してプロセスキュー2に再投入される。従って、次に演算プロセッサにスケジュールされる際には、その時点で最も最適と判断された演算用プロセッサにスケジュールされることとなる。

【0042】これにより、システム全体の負荷バランスを考慮した入出力処理の負荷分散制御が実現され、かつ同一の入出力装置から短期間に大量の入出力処理が発生した場合等の処理効率を損なうこともない。

【0043】図4を参照して第1実施例の動作を説明する。図4は第1実施例の動作を説明するためのフローチャートである。入出力装置の入出力処理完了等に基づく入出力処理要求が発生すると（図4のステップA1）、入出力処理分散部6は、リクエストチケットテーブル7を参照し、リクエストチケットテーブル7に示されたリターンテーブル5a～5bにこの入出力処理を登録する（図4のステップA2）。

【0044】一方、スケジューラ4は、プロセスキュー2に登録されている入出力処理プロセス3a～3bを稼働状況等に応じて演算用プロセッサ1a～1bのいずれかにスケジュールする（図4のステップA3）。

【0045】そして、この演算用プロセッサ1a～1bのいずれかにスケジュールされた入出力処理プロセス3a～3bは、リターンチケットテーブル5a～5bに登録されている入出力処理を実行する（図4のステップA4）。

【0046】これによりシステム全体の負荷バランスを考慮した入出力処理の負荷分散制御が実現される。次に、図5乃至図9を参照して本発明の第2実施例を説明する。

【0047】図5は、第2実施例に係る入出力処理負荷分散制御方式を適用してなるマルチプロセッサシステムのを概略構成図である。同実施例に係るマルチプロセッサシステムは、第1実施例のマルチプロセッサシステムの構成に加え、さらに、平均応答時間テーブル9及び入出力処理回数テーブル10を設ける。

【0048】この平均応答時間テーブル9は、演算用プロセッサ1a~1bそれぞれから渡される各種入出力処理の平均応答時間を格納する。また、入出力処理回数テーブル10は、各入出力装置から要求された入出力処理の回数を入出力装置単位に格納する。

【0049】同実施例では、入出力処理プロセス3a~3bそれぞれを処理する演算用プロセッサ1a~1bを予め特定の演算用プロセッサ1a~1bに連結させておき、スケジューラ4は、プロセスキュー2に蓄えられた入出力処理プロセス3a~3bを、予め設定により指定された演算用プロセッサ1a~1bにスケジュールする。

【0050】いま、例えば入出力装置の入出力処理完了等に基づく入出力処理要求の割り込みが発生したとすると、まず、入出力処理負荷分散部6が、リクエストチケットテーブル7を参照し、この入出力装置の入出力処理をリターンチケットテーブル5a~5bのいずれに登録すべきものなのか、即ち演算用プロセッサ1a~1bのいずれに実行させるべきものなのかを判定する（同実施例では、入出力処理プロセスを実行する演算用プロセッサが静的に連結されているため、リターンチケットテーブルの選択が、そのまま演算用プロセッサの選択となる）。

【0051】この判断により、登録すべきリターンチケットテーブル、即ち、処理を行う演算用プロセッサが判明すると、今度は、平均応答時間テーブルを参照し、この演算用プロセッサの平均応答時間が、他の演算用プロセッサの平均応答時間の平均値に対して所定の基準以上の値になっていないかを判定する。

【0052】そして、この判定の結果、所定の基準以上の値になっていることが判明した場合、さらに、入出力処理回数テーブル10を参照し、この入出力装置の入出力要求が、その演算用プロセッサが処理する入出力装置中、最も回数が多いものであるかを判定する。

【0053】ここで、最も回数が多いと判定された場合、入出力処理負荷分散部6は、この入出力装置の入出力処理を最も平均応答時間の小さい演算用プロセッサに処理させるように、リクエストチケットテーブル7を更新する。

【0054】例えば、リクエストチケットテーブル7が図6に示すように設定されている状態で、入出力装置番号が5の入出力装置からの入出力処理要求が発生した場合、入出力処理負荷分散部6は、リクエストチケットテーブル7を参照することにより、この入出力処理を2番

の演算プロセッサに処理させるものであることを認識する。

【0055】ここで、入出力処理負荷分散部6は、今度は平均応答時間テーブル10を参照し、この2番の演算プロセッサの応答時間が他の演算プロセッサの平均応答時間の平均値に対して所定倍以上の値となっていないかを判定する。

【0056】例えば、平均応答時間テーブル10が図7に示すようになっており、かつ、基準を2倍以内とすると、2番以外の演算プロセッサの平均応答時間の平均値は、 $(1200 + 2400) / 2 = 1800$ となり、2番の演算プロセッサの平均応答時間は、この値の2倍以上となる。

【0057】このとき、入出力処理負荷分散部6は、さらに入出力処理回数テーブル10を参照し、入出力装置番号が5の入出力装置が、2番の演算プロセッサの処理する入出力処理の中で最も多いものであるかを判定する。例えば、入出力処理回数テーブル10が、図8に示す状態である場合、2番の演算プロセッサの処理する入出力処理装置は、リクエストチケットテーブル7から3番と5番であることが分かり、これにより、この5番の入出力処理装置の入出力処理の回数が、2番の演算プロセッサの処理する入出力処理装置の入出力処理中最も多いものであることが判明する。

【0058】このとき、入出力処理負荷分散部6は、この5番の入出力処理装置の入出力処理を最も平均応答時間の小さい演算プロセッサに処理させるように、リクエストチケットテーブル7を更新する。この場合、0番の演算用プロセッサが最も平均応答時間の小さい演算プロセッサになるため、リクエストチケットテーブル7の5番の入出力処理装置の入出力処理を行う演算用プロセッサを2番から0番に更新する。

【0059】この後、入出力負荷分散部6は、リクエストチケットテーブル7に従って、リターンチケットテーブルにこの入出力処理を登録する。このように、各演算用プロセッサの平均応答時間と入出力処理の回数を考慮して、リクエストチケットテーブルを動的に更新していくことにより、システムの稼働状況に応じた入出力処理の負荷分散制御が実現されることとなる。

【0060】図9を参照して第2実施例の動作を説明する。図9は第2実施例の動作を説明するためのフローチャートである。入出力装置の入出力処理完了等に基づく入出力処理要求が発生すると（図9のステップB1）、入出力処理分散部6は、まず入出力処理回数テーブル10のカウントアップを行う（図9のステップB2）。

【0061】次に入出力処理分散部6は、リクエストチケットテーブルを参照し、この入出力処理を行う演算用プロセッサ番号を取得し（図9のステップB3）、さらに平均応答時間テーブル9を参照し、この演算用プロセッサ番号の平均応答時間を取得する（図9のステップB

4)

ここで、入出力処理分散部6は、この平均応答時間が、他の演算用プロセッサの平均応答時間の平均値に対し、所定の基準値以上になっていないかを判定する(図9のステップB5)。

【0062】ここで、基準値以上であると判定された場合には(図9のステップB5のY)、さらに入出力処理の発生回数がこの演算用プロセッサ中最多であるか否かを判定する(図9のステップB6)。

【0063】ここで、最多であると判定された場合には(図9のステップB6のY)、この入出力装置の入出力処理を、最も平均応答時間の小さい演算用プロセッサに処理させるように、リクエストチケットテーブルを更新する(図9のステップB7)。

【0064】この後、入出力処理負荷分散部6は、リクエストチケットテーブルにより指定されたリターンテーブルに、この入出力処理を登録する(図9のステップB8)。

【0065】一方、スケジューラ4は、プロセスキュー2に登録されている入出力処理プロセスを予め設定により指定された演算用プロセッサにスケジュールする(図9のステップB9)。そして、この予め設定により指定された演算用プロセッサにスケジュールされた入出力処理プロセスは、リターンチケットテーブルに登録されている入出力処理を実行する(図9のステップB10)。

【0066】これにより、スケジューラのスケジュール機能によらずにシステム全体の負荷バランスを考慮した入出力処理の負荷分散制御が実現される。なお、図10に示すように、さらに装置設置検出部11を設け、マルチプロセッサシステムの稼働中における演算用プロセッサ及び入出力装置等の増減を含むシステム構成の変更を検知し、この検知結果に従って、入出力処理プロセス、リクエストチケットテーブル、リターンチケットテーブル、平均応答時間テーブル、及び入出力処理回数テーブルの増減を動的に実施させれば、システム稼働中にシステム構成が変更された際においても、システムの停止及び再起動を実施することなく最適な負荷分散を継続して行うことを可能とする。

【0067】

【発明の効果】以上詳述したように本発明によれば、処理効率を損なうことなくシステム全体の負荷バランスを考慮した入出力処理の負荷分散制御が実現され、また、システム稼働中にシステム構成が変更された際においても、システムの停止及び再起動を実施することなく最適な負荷分散を継続して行うことを可能とする。

【図面の簡単な説明】

【図1】本発明の第1実施例に係る入出力処理負荷分散制御方式を適用してなるマルチプロセッサシステムの概略構成図。

【図2】第1実施例に係るリクエストチケットテーブルを示す概念図。

【図3】第1実施例に係るリターンチケットテーブルを示す概念図。

【図4】第1実施例の動作を説明するためのフローチャート。

【図5】本発明の第2実施例に係る入出力処理負荷分散制御方式を適用してなるマルチプロセッサシステムの概略構成図。

【図6】第2実施例に係るリクエストチケットテーブルを示す概念図。

【図7】第2実施例に係る平均応答時間テーブルを示す概念図。

【図8】第2実施例に係る入出力処理回数テーブルを示す概念図。

【図9】第2実施例の動作を説明するためのフローチャート。

【図10】装置設置検出手段を備えた入出力処理負荷分散制御方式を適用してなるマルチプロセッサシステムの概略構成図。

【図11】従来のマルチプロセッサシステムでの入出力処理の動作とその問題点を示す図。

【符号の説明】

1a、1b…演算用プロセッサ、2…プロセスキュー、3a、3b…入出力処理プロセス、4…スケジューラ、5a、5b…リターンチケットテーブル、6…入出力処理分散部、7…リクエストチケットテーブル、8…各種入出力処理、9…平均応答時間、10…入出力処理回数テーブル、11…装置設置検知部。

【図2】

【図3】

【図6】

【図8】

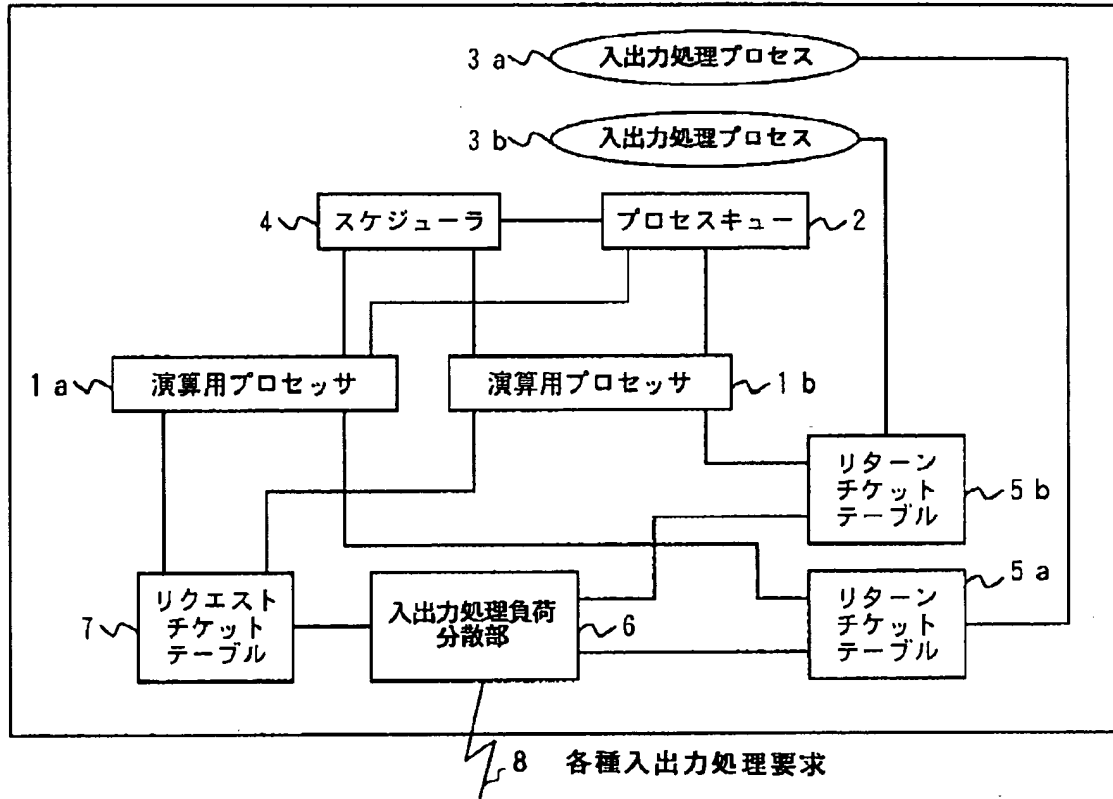
入出力装置番号	リターンチケット テーブル
1	0
2	1
3	0

入出力装置番号
1
3
⋮

入出力装置	演算用プロセッサ番号
1	0
2	1
3	2
4	0
5	2

入出力装置	入出力処理回数
1	112
2	5
3	500
4	201
5	798

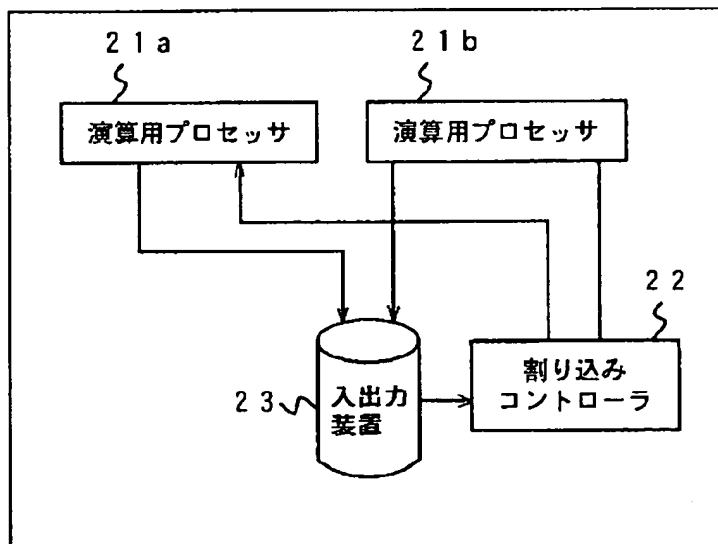
【図1】



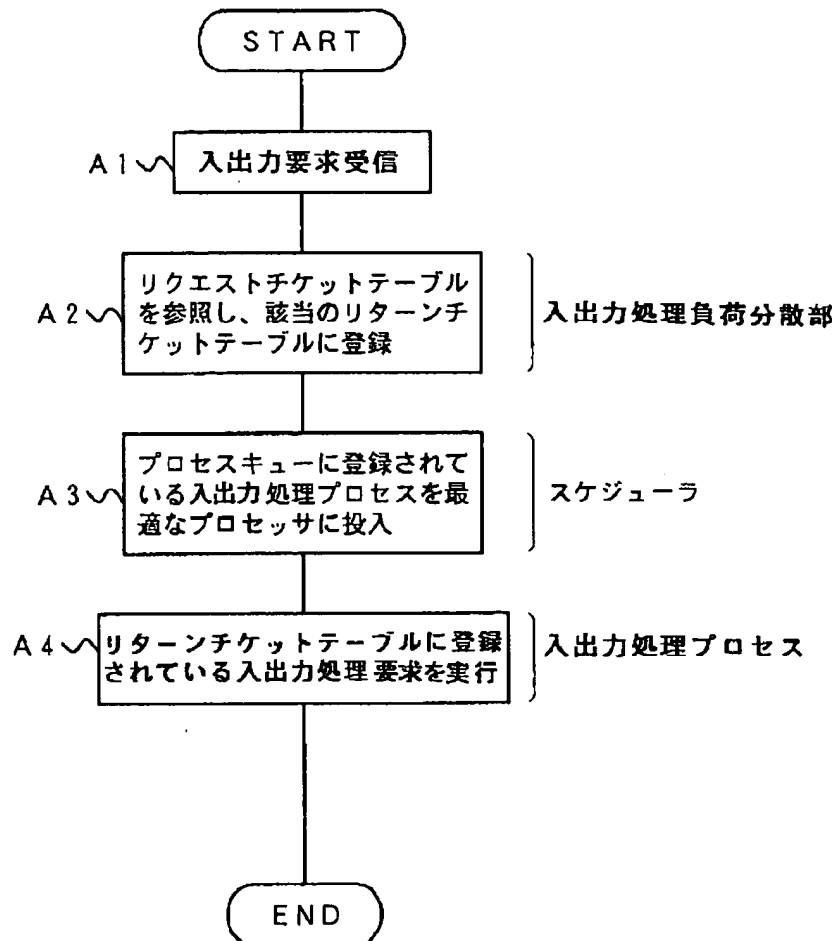
【図7】

演算用プロセッサ	平均応答時間
0	1200
1	2400
2	9000

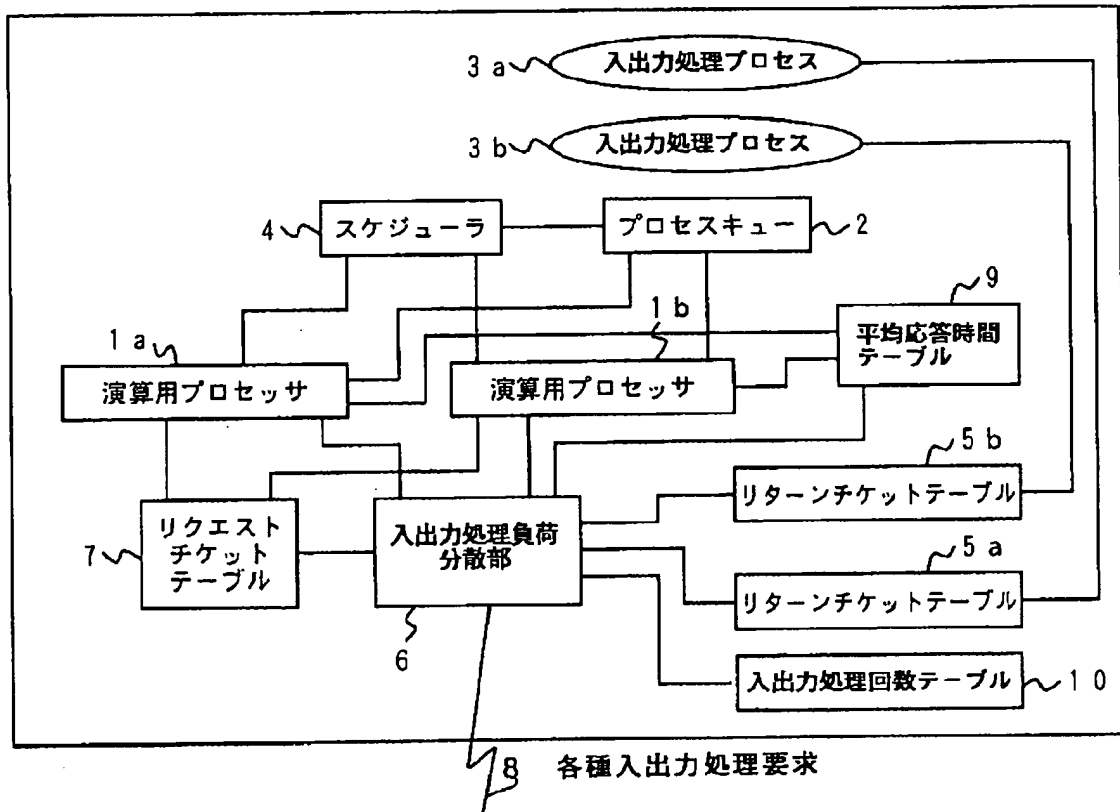
【図11】



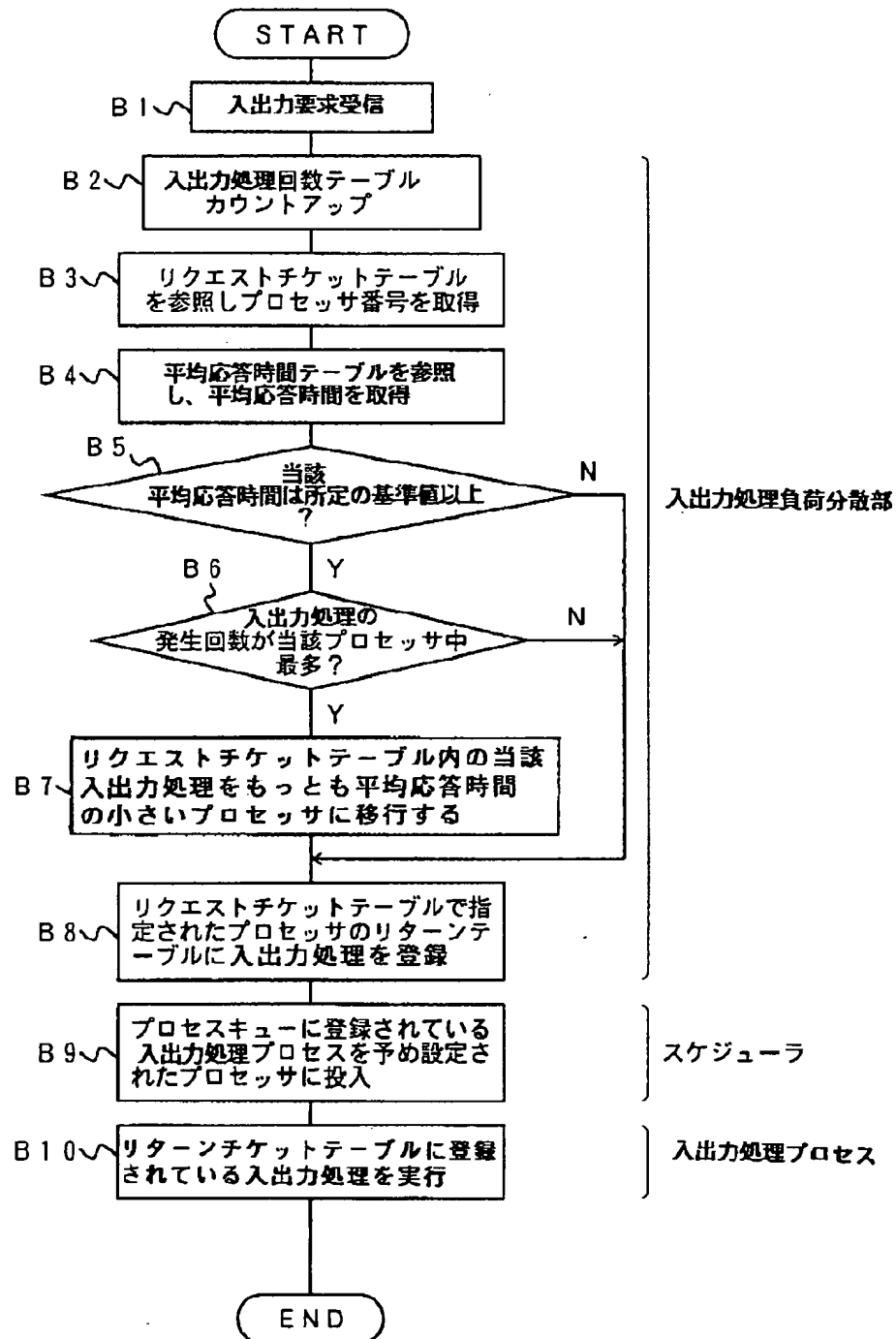
【図4】



【図5】



【図9】



【図10】

